FACTORY PHYSICS

Foundations of Manufacturing Management

SECOND EDITION

Wallace J. Hopp
Northwestern University

Mark L. Spearman
Georgia Institute of Technology

Irwin
McGraw-Hill

Boston Burr Ridge, IL Dubuque, IA Madison, WI New York San Francisco St. Louis
Bangkok Bogotá Caracas Lisbon London Madrid
Mexico City Milan New Delhi Seoul Singapore Sydney Taipei Toronto

CHAPTER

7 BASIC FACTORY DYNAMICS

I do not know what I may appear to the world; but to myself I seem to have been only like a boy playing on the seashore, and diverting myself in now and then finding a smoother pebble or a prettier shell than ordinary, whilst the great ocean of truth lay all undiscovered before me.

Isaac Newton

7.1 Introduction

In the previous chapter, we argued that manufacturing management needs a science of manufacturing. In this chapter, we begin the process of fleshing out such a science by examining some basic behavior of production lines.

To motivate the measures and mechanics on which we will focus, we begin with a realistic example. HAL, a computer company, manufactures printed-circuit boards (PCBs), which are sold to other plants, where the boards are populated with components ("stuffed") and then sent to be used in the assembly of personal computers. The basic processes used to manufacture PCBs are as follows:

1. *Lamination.* Layers of copper and prepreg (woven fiberglass cloth impregnated with epoxy) are pressed together to form cores (blank boards).
2. *Machining.* The cores are trimmed to size.
3. *Circuitize.* Through a photographic exposing and subsequent etching process, circuitry is produced in the copper layers of the blanks, giving the cores "personality" (i.e., a unique product character). They are now called *panels.*
4. *Optical test and repair.* The circuitry is scanned optically for defects, which are repaired if not too severe.
5. *Drilling.* Holes are drilled in the panels to connect circuitry on different planes of multilayer boards. Note that multilayer panels must return to lamination after being circuitized to build up the layers. Single-layer panels go through lamination only once and do not require drilling or copper plating.
6. *Copper plate.* Multilayer panels are run through a copper plating bath, which deposits copper inside the drilled holes, thereby connecting the circuits on different planes.

7. *Procoat.* A protective plastic coating is applied to the panels.
8. *Sizing.* Panels are cut to final size. In most cases, multiple PCBs are manufactured on a single panel and are cut into individual boards at the sizing step. Depending on the size of the board, there could be as few as two boards made from a panel, or as many as 20.
9. *End-of-line test.* An electrical test of each board's functionality is performed.

HAL engineers monitor the capacity and performance of the PCB line. Their best estimates of capacity are summarized in Table 7.1, which gives the average process rate (number of panels per hour) and average process time (hours) at each station. (Note that because panels are often processed in batches and because many processes have parallel machines, the rate of a process is not the inverse of the time.) These values are averages, which account for the different types of PCBs manufactured by HAL and also the different routings (e.g., some panels may visit lamination twice). They also account for "detractors," such as machine failures, setup times, and operator efficiency. As such, the process rate gives an approximation of how many panels each process could produce per hour if it had unlimited inputs. The process time represents the average time a typical panel spends being worked on at a process, which includes time waiting for detractors but *does not* include time waiting in queue to be worked on.

The main performance measures emphasized by HAL are throughput (how many PCBs are produced), cycle time (the time it takes to produce a typical PCB), work in process (inventory in the line), and customer service (fraction of orders delivered to customers on time). Over the past several months, throughput has averaged about 1,100 panels per day, or about 45.8 panels per hour (HAL works a 24-hours a day). WIP in the line has averaged about 37,000 panels, and manufacturing cycle time has been roughly 34 days, or 816 hours. Customer service has averaged about 75 percent.

The question is, how is HAL doing?

We can answer part of this question immediately. HAL management is not happy with 75 percent customer service because it has a corporate goal of 90 percent. So this aspect of performance is not good. However, perhaps the reason for this is that overzealous salespersons are promising unrealistic due dates to customers. It may not be an indication of anything wrong with the line.

The other measures—throughput, WIP and cycle time—are more difficult to deal with. We need to establish some sort of baseline against which to compare them. One

TABLE 7.1 Capacity Data for HAL Printed-Circuit Board Line

Process	Rate (parts per hour)	Time (hour)
Lamination	191.5	1.2
Machining	186.2	5.9
Circuitize	150.5	6.9
Optical test/repair	157.8	5.6
Drilling	185.9	10.0
Copper plate	136.4	1.5
Procoat	146.2	2.2
Sizing	126.5	2.4
EOL test	169.5	1.8

way to do this would be to benchmark against a competitor's operation. But even if HAL could get such data, there would still be the question of how comparable they really were. After all, every facility is unique. To be better or worse than a different type of facility does not necessarily mean much. A better baseline would be one that compares actual performance against what is theoretically possible for this facility.

In this chapter, we examine the extremes of behavior that are possible for simple idealized production lines, and we use the resulting models to develop a scale with which to rate actual facilities. We will return to the HAL example and use this scale to evaluate the performance of its PCB line. But first we must define our terms.

7.2 Definitions and Parameters

The scientific method absolutely requires precise terminology. Unfortunately, use of manufacturing terms in industry and the OM literature is far from standardized. This can make it extremely difficult for managers and engineers from different companies (and even the same company) to communicate and learn from one another. What it means for us is that the best we can do is to define our terms carefully and warn the reader that other sources will use the same terms differently or use different terms in place of ours.

7.2.1 Definitions

In Part II, we focus on the behavior of production *lines,* because these are the links between individual processes and the overall plant. Therefore, the following terms are defined in a manner that allows us to describe lines with precision. Some of these terms also have broader meanings when applied to the plant, as we note in our definitions and will occasionally adopt in Part III. However, to develop sharp intuition about production lines, we will maintain these rather narrow definitions for the remainder of Part II.

Workstation: A **workstation** is a collection of one or more machines or manual stations that perform (essentially) identical functions. Examples include a turning station made up of several vertical lathes, an inspection station made up of several benches staffed by quality inspectors, and a burn-in station consisting of a single room where components are heated for testing purposes. In **process-oriented layouts,** workstations are physically organized according to the operations they perform (e.g., all grinding machines located in the grinding department). Alternatively, in **product-oriented layouts** they are organized in lines making specific products (e.g., a single grinding machine dedicated to an individual line). The terms **station, workcenter,** and **process center** are synonymous with *workstation.*

Part: A **part** is a piece of raw material, a component, a subassembly, or an assembly that is worked on at the workstations in a plant. **Raw material** refers to parts purchased from outside the plant (e.g., bar stock). **Components** are individual pieces that are assembled into more complex products (e.g., gears). **Subassemblies** are assembled units that are further assembled into more complex products (e.g., transmissions). **Assemblies** (or final assemblies) are fully assembled products or end items (e.g., automobiles). Note that one plant's final assemblies may be another's raw material. For instance, transmissions are the final assemblies of a transmission plant, but are raw materials or purchased components to the automotive assembly plant.

End item: A part that is sold directly to a customer, whether or not it is an assembly, is called an **end item.** The relationship between end items and their constituent parts

(raw materials, components, and subassemblies) is maintained in the **bill of material (BOM),** which Chapter 3 presented in detail.

Consumable: For the most part, **consumables** are materials such as bits, chemicals, gases, and lubricants that are used at workstations but do not become part of a product that is sold. More formally, we distinguish between parts and consumables in that parts are listed on the bill of material, while consumables are not. This means that some items that do become part of the product, such as solder, glue, and wire, can be considered either parts if they are recorded on the bill of material or consumables if they are not. Since different purchasing schemes are typically used for parts and consumables (e.g., parts might be ordered according to an MRP system, while consumables are purchased through a reorder point system), this choice may influence how such items are managed.

Routing: A **routing** describes the sequence of workstations passed through by a part. Routings begin at a raw material, component, or subassembly stock point and end at either an intermediate stock point or finished-goods inventory. For instance, a routing for gears may start at a stock point of raw bar stock; pass through cutting, hobbing, and deburring; and end at a stock point of finished gears. This stock of gears might in turn feed another routing that builds gear subassemblies. The bill of material and the associated routings contain the basic information needed to make an end item.

Order: A **customer order** is a request from a customer for a particular part number, in a particular quantity, to be delivered on a particular date. The paper or electronic **purchase order** sent by the customer might contain several customer orders. Henceforth, we will refer to a customer order as simply an **order.** Inside the plant, an order can also be an indication that certain inventories (e.g., safety stocks) need to be replenished. While timing may be more critical for orders originating with customers, both types of orders represent demand.

Job: A **job** refers to a set of physical materials that traverses a routing, along with the associated logical information (e.g., drawings, BOM). Although every job is triggered by either an actual customer order or the anticipation of a customer order (e.g., forecasted demand), there is frequently not a one-to-one correspondence between jobs and orders. This is because (1) jobs are measured in terms of specific parts (uniquely identified by a part number), not the collection of parts that may make up the assembly required to satisfy an order, and (2) the number of parts in a job may depend on manufacturing efficiency considerations (e.g., batch size considerations) and thus may not match the quantities ordered by customers.

Throughput (TH): The average output of a production process (machine, workstation, line, plant) per unit time (e.g., parts per hour) is defined as the system's **throughput,** or sometimes **throughput rate.** At the firm level, throughput is defined as the production per unit time that is *sold.* However, managers of production lines generally control what is made rather than what is sold. Therefore, for a plant, line, or workstation, we define throughput to be the average quantity of *good* (nondefective) parts (the manager does have control over quality) produced per unit time. In a line made up of workstations in tandem dedicated to a single family of products and where all products pass through each station exactly once, the throughput at every station will be the same (provided there is no yield loss). In a more complex plant, where workstations service multiple routings (e.g., a job shop), the throughput of an individual station will be the sum of the throughputs of the routings passing through it.

Capacity: An upper limit on the throughput of a production process is its **capacity.** In most cases, releasing work into the system at or above the capacity causes the system to become unstable (i.e., build up WIP without bound). Only very special systems can operate stably at capacity. Because this concept is subtle and important, we will inves-

tigate it more thoroughly later in this chapter, once we have introduced the appropriate notation and concepts.

Raw material inventory (RMI): As noted, the physical inputs at the start of a production process are typically called **raw material inventory.** This could represent bar stock that is cut up and then milled into gears, sheets of copper and fiberglass that are laminated together to make circuit boards, wood chips that become pulp and then paper stock, or rolls of sheet steel that are pressed into automobile fenders. Typically, the stock point at the beginning of a routing is termed raw material inventory even though the material may have already undergone some processing.

"Crib" and finished goods inventory (FGI): The stock point at the end of a routing is either a **crib inventory location** (i.e., an intermediate inventory location) or **finished goods inventory.** Crib inventories are used to gather different parts within the plant before further processing or assembly. For instance, a routing to produce gear assemblies may be fed by several crib inventories containing gears, housings, crankshafts, and so on. Finished goods inventory is where end items are held prior to shipping to the customer.

Work in process (WIP): The inventory between the start and end points of a product routing is called **work in process (WIP).** Since routings begin and end at stock points, WIP is all the product between, but not including, the ending stock points. Although in colloquial use WIP often includes crib inventories, we make a distinction between crib inventory and WIP to help clarify the discussion.

Inventory turns: A commonly used measure of the efficiency with which inventory is used is **inventory turns,** or the **turnover ratio,** which is defined as the ratio of throughput to average inventory. Typically, throughput is stated in yearly terms, so that this ratio represents the average number of times the inventory stock is replenished or turned over. Exactly which inventory is included depends on what is being measured. For instance, in a warehouse, all inventory is FGI, so turns are given by TH/FGI. In a plant, we generally consider both WIP (inventory still in the line) and FGI (inventory waiting to ship), so turns are given by TH/(WIP + FGI). In any case, it is essential to make sure that throughput and inventory are measured in the same units. Since inventory is usually measured in cost dollars (i.e., rather than price or sales dollars), throughput should also be measured in cost dollars.

Cycle time (CT): The **cycle time** (also called variously **average cycle time, flow time, throughput time,** and **sojourn time**) of a given routing is the average time from release of a job at the beginning of the routing until it reaches an inventory point at the end of the routing (i.e., the time the part spends as WIP).[1] Although this is a precise definition of cycle time, it is also narrow, allowing us to define cycle time only for individual routings. It is common for people to refer to the cycle time of a product that is composed of many complex subassemblies (e.g., automobiles). However, it is not clear exactly what is meant by this. When does the clock start for an automobile? When the chassis starts down the assembly line? When the engine begins production? Or, as in Henry Ford's terms, when the ore is mined from the ground? We will discuss measuring cycle time for such assembled parts later, but for now we restrict our definition to single routings.

Lead time, service level, and fill rate: The **lead time** of a given routing or line is the time allotted for production of a part on that routing or line. As such, it is a management constant.[2] In contrast, cycle times are generally random. Therefore, in a line functioning

[1] Cycle time also has another meaning in assembly lines as the time allotted for each station to complete its task. It can also refer to the processing time of an individual machine (e.g., the time for a punch press to cycle). We will avoid these other uses of the term *cycle time* to prevent confusion.

[2] Recall that the time phasing function of MRP is critically dependent on the choice of such lead times.

in a *make-to-order* environment (i.e., it produces parts to satisfy orders with specific due dates), an important measure of line performance is **service level,** which is defined as

$$\text{Service level} = P\{\text{cycle time} \leq \text{lead time}\}$$

Notice that this definition implies that for a given distribution of cycle time, service level can be influenced by manipulating lead time (i.e., the higher the lead time, the higher the service level).

If the line is functioning in a *make-to-stock* environment (i.e., it fills a buffer from which customers or other lines expect to be able to obtain parts without delay), then a different performance measure may be more appropriate than service level. A logical choice is **fill rate,** which is defined as the fraction of orders that are filled from stock and was discussed in Chapter 2. Since fill rate and many other performance measures are often referred to as "service levels," the reader is cautioned to look for a precise definition whenever this term is encountered. We will consistently use the former definition of service level throughout Part II, but will return to the fill rate measure in Chapter 17.

Utilization: The **utilization** of a workstation is the fraction of time it is not idle for lack of parts. This includes the fraction of time the workstation is working on parts or has parts waiting and is unable to work on them due to a machine failure, setup, or other detractor. We can compute utilization as

$$\text{Utilization} = \frac{\text{Arrival rate}}{\text{Effective production rate}}$$

where the effective production rate is defined as the maximum average rate at which the workstation can process parts, considering the effects of failures, setups, and all other detractors that are relevant over the planning period of interest.

7.2.2 Parameters

Parameters are numerical descriptors of manufacturing processes and therefore vary in value from plant to plant. Two key parameters for describing an individual line (routing) are the bottleneck rate and the raw process time. We define these below, along with a third parameter, the *critical* WIP level, that can be computed from them.

Bottleneck rate (r_b): The **bottleneck rate** of the line, r_b, is the rate (parts per unit time or jobs per unit time) of the workstation having the highest long-term utilization. By long term we mean that outages due to machine failures, operator breaks, quality problems, etc., are averaged out over the time horizon under consideration. This implies that the proper treatment of outages will differ depending on the planning frequency. For example, for daily replanning, outages typically experienced during a day should be included; but unplanned long outages, such as those resulting from a major upset, should not. In contrast, for planning over a year-long horizon, time lost to major upsets should be included, if such occurrences are not unlikely over the course of a year.

In lines consisting of a single routing in which each station is visited exactly once and there is no yield loss, the arrival rate to every workstation is the same. Hence, the workstation with the highest utilization will be that with the least long-term capacity (i.e., slowest effective rate). However, in lines with more complicated routings or yield loss, the bottleneck may not be at the slowest workstation. A faster workstation that experiences a higher arrival rate may have higher utilization. For this reason, it is important to define the bottleneck in terms of utilization as we have done here.

Raw process time (T_0): The **raw process time** of the line, T_0, is the sum of the *long-term average* process times of each workstation in the line. Alternatively, we can

define raw process time as the average time it takes a single job to traverse the empty line (i.e., so it does not have to wait behind other jobs). Again, we must be concerned about the length of the planning horizon when deciding what to include in the "average" process times. Over the long term, T_0 should include infrequent random and planned outages, while over a shorter term it should include only the more frequent delays.

Critical WIP (W_0): The **critical WIP** of the line, W_0, is the WIP level for which a line with given values of r_b and T_0 but having no variability achieves maximum throughput (that is, r_b) with minimum cycle time (that is, T_0). We show below that critical WIP is defined by the bottleneck rate and raw process time by the following relationship:

$$W_0 = r_b T_0$$

7.2.3 Examples

We now illustrate these definitions by means of two simple examples.

Penny Fab One. Penny Fab One consists of a simple production line that makes giant one-cent pieces used exclusively in Fourth of July parades. The line consists of four machines in sequence that use well-known, stable processes. The first machine is a punch press that cuts penny blanks, the second stamps Lincoln's face on one side and the Memorial on the back, the third places a rim on the penny, and the fourth cleans away any burrs. Each machine takes exactly two hours to perform its operation. (We will relax this requirement that process times be deterministic later.) After each penny is processed, it is moved immediately to the next machine. The line runs 24 hours per day, with breaks, lunches, etc., covered by spare operators. For our purposes, the market for giant pennies can be assumed to be unlimited, so that all product made is sold; thus, more throughput is unambiguously better for this system.

Since this is a tandem line with no yield loss, the bottleneck is defined as the slowest workstation. However, the *capacity* of each machine is the same and equals one penny every two hours, or one-half part per hour. Hence, any of the four machines can be regarded as the bottleneck and

$$r_b = 0.5 \text{ penny per hour}$$

or 12 pennies per day. Such a line is said to be **balanced,** since all stations have equal capacity.

Next, note that the raw process time is simply the sum of the processing times at the four stations, so

$$T_0 = 8 \text{ hours}$$

The critical WIP level is given by

$$W_0 = r_b T_0 = 0.5 \times 8 = 4 \text{ pennies}$$

We will illustrate that this is indeed the level of WIP that causes the line to achieve throughput of $r_b = 0.5$ penny per hour and cycle time of $T_0 = 8$ hours. Notice that W_0 is equal to the number of machines in the line. This is *always* the case for balanced lines, since having one job per machine is just enough to keep all machines busy at all times. However, as we will see, it is not true for unbalanced lines.

Penny Fab Two. Now consider a somewhat more complex Penny Fab Two, which represents an unbalanced line with multimachine stations. Penny Fab Two still produces giant pennies in four steps: punching, stamping, rimming, and deburring; but the

workstations now have different numbers of machines and processing times, as shown in Table 7.2.

The presence of multimachine stations complicates the capacity calculations somewhat. For a single machine, the capacity is simply the reciprocal of the process time (e.g., if it takes one-half hour to do one job, the machine can do two jobs per hour). The capacity of a station consisting of several identical machines in parallel must be calculated as the individual machine capacity times the number of machines. For example, in Penny Fab Two, the capacity per machine at station 3 is

$$\frac{1}{10} \text{ penny per hour}$$

so the capacity of the station is

$$6 \times \frac{1}{10} = 0.6 \text{ penny per hour}$$

Notice that the station capacity can be computed directly by dividing the number of machines by the process time. This is done for each station in Table 7.2.

The capacity of the line with multimachine stations is still defined by the rate of the bottleneck, or slowest station in the line. In Penny Fab Two, the bottleneck is station 2, so

$$r_b = 0.4 \text{ penny per hour}$$

Notice that the bottleneck is neither the station that contains the slowest machines (station 3) nor the one with the fewest machines (station 1).

The raw process time of the line is still the sum of the process times. Notice that adding machines at a station does not decrease T_0, since a penny can be worked on by only one machine at a time. Hence, the raw process time for Penny Fab Two is

$$T_0 = 20 \text{ hours}$$

Regardless of whether the line has single- or multiple machine stations, the critical WIP level is always defined as

$$W_0 = r_b T_0 = 0.4 \times 20 = 8 \text{ pennies}$$

In Penny Fab Two, as in Penny Fab One, W_0 is a whole number. This, of course, need not be the case. If W_0 comes out to a fraction, it means that there is no constant WIP level that will achieve throughput of exactly r_b jobs per hour and cycle time of T_0 hours. Furthermore, notice that the critical WIP level in Penny Fab Two (eight pennies) is less than the number of machines (11). This is because the system is not balanced (i.e., stations have different amounts of capacity), and therefore some stations will not be fully utilized.

TABLE 7.2 Penny Fab Two: An Unbalanced Line

Station Number	Number of Machines	Process Time (hour)	Station Capacity (Jobs per Hour)
1	1	2	0.50
2	2	5	0.40
3	6	10	0.60
4	2	3	0.67

7.3 Simple Relationships

Now, in the pursuit of a science of manufacturing, we ask the fundamental question, What are the relationships among WIP, throughput, and cycle time in a single production line? Of course, the answer will depend on the assumptions we make about the line. In this section, we will give a precise (i.e., quantitative) description of the range of possible behavior. This will serve to sharpen our intuition about how lines perform and will give us a scale on which to rate (benchmark) actual systems.

A problem with characterizing the relationship between measures such as WIP and throughput is that in real systems they tend to vary simultaneously. For instance, in an MRP system, the line may be flooded with work one month (due to a heavy master production schedule) and very lightly loaded the next. Hence, both WIP and throughput are apt to be high during the first month and low during the second. For clarity of presentation, we will eliminate this problem by controlling the WIP level in the line so as to hold it constant over time. For instance, in the Penny Fabs, we will start the lines with a specified number of pennies (jobs) and then release a new penny blank into the line each time a finished penny exits the line.[3]

7.3.1 Best-Case Performance

To analyze and understand the behavior of a line under the best possible circumstances, namely, when process times are absolutely regular, we will *simulate* Penny Fab One. This is easily done by using a piece of paper and several pennies, as shown in Figure 7.1.

We begin by simulating the system when only one job is allowed in the line. The first penny spends two hours successively at stations 1, 2, 3, and 4, for a total cycle time of eight hours. Then a second penny is released into the line, and the same sequence is repeated.

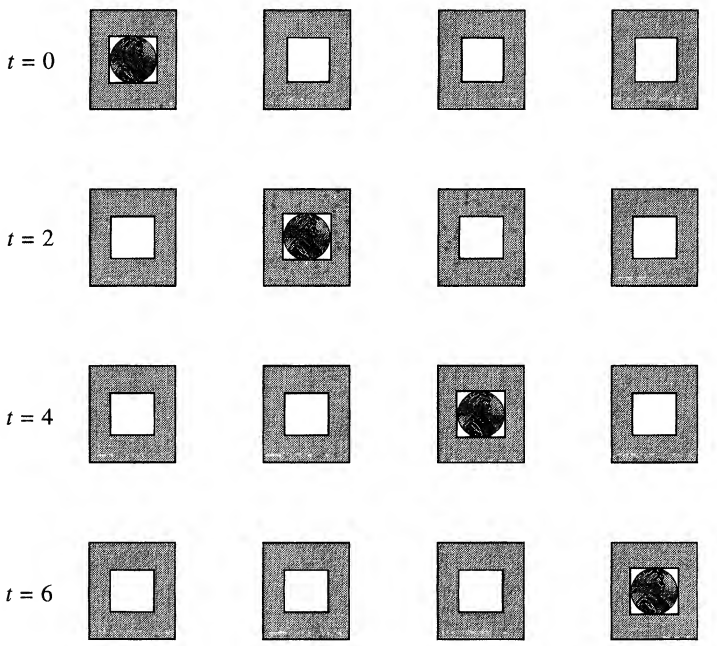


FIGURE 7.1
Penny Fab One with WIP = 1

[3] We say that such a line is operating under a CONWIP (*con*stant *WIP*) protocol, which is treated more thoroughly in Chapters 10 and 14.

Since this results in one penny coming out of the line every eight hours, the throughpu is one-eighth penny per hour. Notice that the cycle time is equal to the raw process time $T_0 = 8$, while the throughput is one-fourth of the bottleneck rate $r_b = 0.5$.

Now we add a second penny to the line (starting both at the front of the line). After two hours, the first penny completes processing at station 1 and starts on station 2. Simultaneously, the second penny starts processing on station 1. Thereafter, the second penny will follow the first, switching stations every two hours, as shown in Figure 7.2. After the initial wait experienced by the second penny, it never waits again. Hence, once the system is running in steady state, every penny released into the line still has a cycle time of exactly eight hours. Moreover, since two pennies exit the line every eight hours, the throughput increases to two-eighths penny per hour, double that when the WIP level was 1 and 50 percent of line capacity ($r_b = 0.5$).

We add a third penny. Again, after an initial transient period in which pennies wait at the first station, there is no waiting, as shown in Figure 7.3. Hence, cycle time stays at 8 h, while throughput increases to three-eighths part per hour, or 75 percent of r_b.

When we add a fourth penny, we see that all the stations stay busy all the time once steady state has been reached. Because there is no waiting at the stations, cycle time is still $T_0 = 8$ h. Since the last station is busy all the time, it outputs a penny every other hour, so throughput becomes one-half penny per hour, which equals the line capacity r_b. This very special behavior, in which cycle time T_0 (its minimum value) and throughput r_b (its maximum value) are only achieved when the WIP level is set at the critical WIP level, which we recall for Penny Fab One is

$$W_0 = r_b T_0 = 0.5 \times 8 = 4 \text{ pennies}$$

Now we add a fifth penny to the line. Because there are only four machines, a penny will wait at the first station, even after the system has settled into steady state. Since we measure cycle time as the time from when a job is released (the time it enters the queue at the first station) to when it exits the line, it now becomes 10 hours, due to the extra two hours of waiting time in front of station 1. Hence, for the first time, cycle time becomes larger than its minimal value $T_0 = 8$. However, since all stations are always busy, the throughput remains at $r_b = 0.5$ penny per hour.

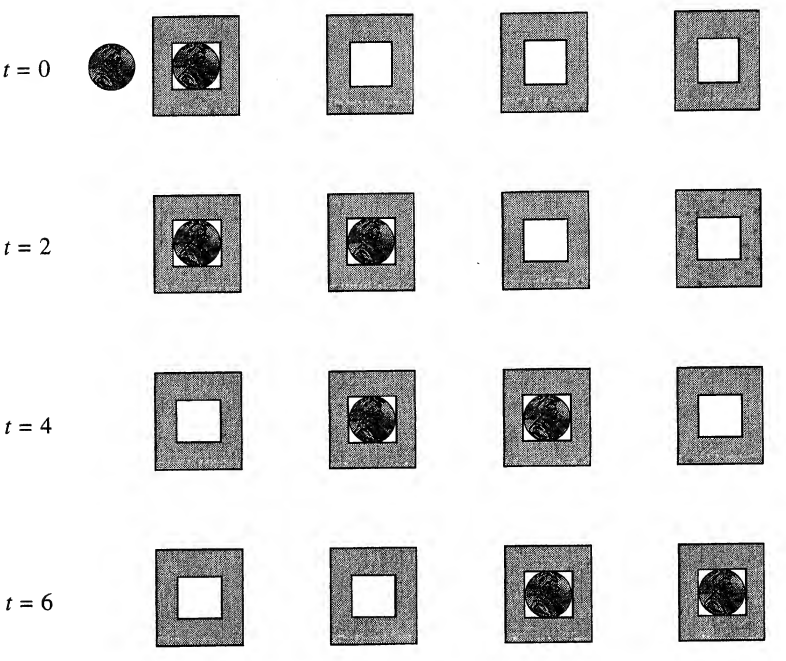


FIGURE 7.2
Penny Fab One with WIP = 2

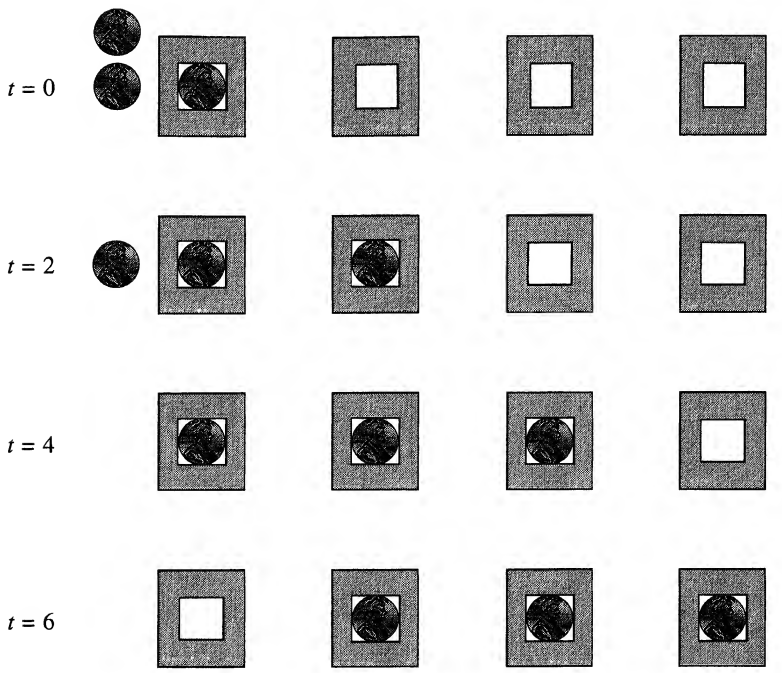


FIGURE 7.3
Penny Fab One with WIP = 3

Finally, consider what happens when we allow 10 pennies in the line. In steady state, a queue of six pennies persists in front of the first station, meaning that an individual penny spends 12 hours from the time it is released to the line until it begins processing at station 1. Hence, the cycle time is 20 hours. As before, all machines remain busy all the time, so throughput is still $r_b = 0.5$ penny per hour. It should be clear at this point that each penny we add increases cycle time by two hours with no increase in throughput.

We summarize the behavior of Penny Fab One with no variability for various WIP levels in Table 7.3, and we present the results graphically in Figure 7.4. From a performance standpoint, it is clear that Penny Fab One runs best when there are four pennies in WIP. Only this WIP level results in minimum cycle time T_0 and maximum throughput r_b—any less and we lose throughput with no decrease in cycle time; any more and we increase cycle time with no increase in throughput. This special WIP level is the critical WIP (W_0) that was defined previously.

In this particular example, the critical WIP is equal to the number of machines. This is always the case when the line consists of stations with equal capacity (i.e., a balanced line). For unbalanced lines, W_0 will be less than the number of machines, but still has the property of being the WIP level that achieves maximum throughput with minimum cycle time, and is still defined by $W_0 = r_b T_0$.

It is important to note that while the critical WIP is optimal in the case with zero variability, it will *not* be optimal in other cases. Indeed, the concept of an optimal WIP level is not even well defined in the presence of variability because, in general, increasing WIP will increase both throughput (good) and cycle time (bad).

Little's Law. Close examination of Table 7.3 reveals an interesting, and fundamental, relationship among WIP, cycle time, and throughput. At every WIP level, WIP is equal to the product of throughput and cycle time. This relation is known as *Little's law* (named for John D. C. Little, who provided the mathematical proof) and represents our first *factory physics* relationship:

Law (Little's Law):

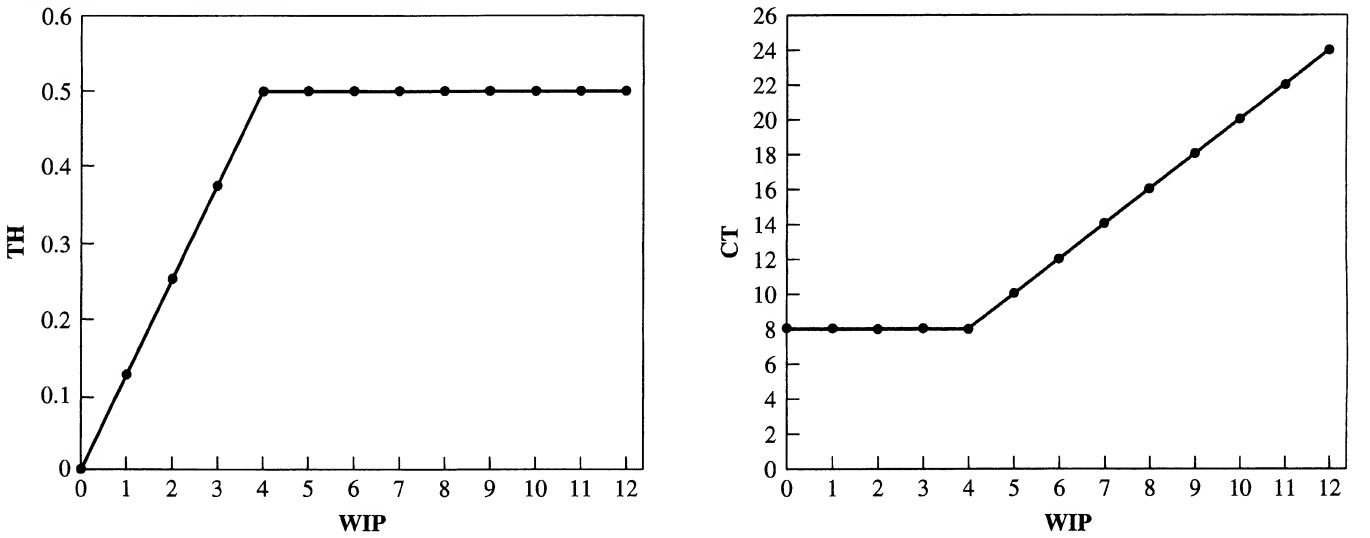
$$\text{WIP} = \text{TH} \times \text{CT}$$

TABLE 7.3 WIP, Cycle Time, and Throughput of Penny Fab One

WIP	CT	% T_0	TH	% r_b
1	8	100	0.125	25
2	8	100	0.250	50
3	8	100	0.375	75
4	8	100	0.500	100
5	10	125	0.500	100
6	12	150	0.500	100
7	14	175	0.500	100
8	16	200	0.500	100
9	18	225	0.500	100
10	20	250	0.500	100

FIGURE 7.4

Cycle time and throughput versus WIP for Penny Fab One



It turns out that Little's law holds for *all* production lines, not just those with zero variability. As we discussed in Chapter 6, Little's law is not a *law* at all but a *tautology.* For special cases (e.g., the case of observing the system for a time that goes to infinity), the relationship can be proved mathematically. However, it does not entirely hold in the less-than-infinite case (which, of course, involves all real cases) except for other special cases. Nonetheless, we will use it as a conjecture about the nature of manufacturing systems and use it as an approximation when it is not exact.

Little's law is quite useful in that it can be applied to a single station, a line, or an entire plant. As long as the three quantities are measured in consistent units, the above relationship will hold over the long term. This makes it immensely applicable to practical situations. Some straightforward uses of Little's law include these:

1. *Queue length calculations.* Since Little's law applies to individual stations, we can use it to calculate the expected queue length and utilization (fraction of time busy) at each station in a line. For instance, consider Penny Fab Two, which was summarized in Table 7.2, and suppose it is running at the bottleneck rate (that is, 0.4 job per hour). From Little's law, the expected WIP at the first station will be

$$\text{WIP} = \text{TH} \times \text{CT} = 0.4 \text{ job per hour} \times 2 \text{ hour} = 0.8 \text{ job}$$

Since there is only one machine at station 1, this means that it will be utilized 80 percent of the time. Similarly, at station 3, Little's law predicts an average WIP of four jobs. Since there are six machines, the average utilization will be $4/6 = 66.7$ percent. Notice that this is equal to the ratio of the rate of the bottleneck to the rate of station 3 (that is, 0.4/0.6), as we would expect.

2. *Cycle time reduction.* Since Little's law can be written as

$$\text{CT} = \frac{\text{WIP}}{\text{TH}}$$

it is clear that reducing cycle time implies reducing WIP, provided throughput remains constant. Hence, large queues are an indication of opportunities for reducing cycle time, as well as WIP. We will discuss specific measures for WIP and cycle time reduction in Chapter 17.

3. *Measure of cycle time.* Measuring cycle time directly can sometimes be difficult, since it entails registering the entry and exit times of each part in the system. Since throughput and WIP are routinely tracked, it might be easier to use the ratio WIP/TH as a perfectly reasonable indirect measure of cycle time.

4. *Planned inventory.* In many systems, jobs are scheduled to finish ahead of their due dates in order to ensure a high level of customer service. Because, in our era of inventory consciousness, customers often refuse to accept early deliveries, this type of "safety lead time" causes jobs to wait in finished goods inventory prior to shipping. If the **planned inventory** time is n days, then according to Little's law, the amount of inventory in FGI will be given by nTH (where TH is measured in units per day).

5. *Inventory turns.* Recall that inventory turns are given by the ratio of throughput to average inventory. If we have a plant in which all inventory is WIP (i.e., product is shipped directly from the line so there is no finished goods inventory), then turns are given by TH/WIP, which by Little's law is simply 1/CT. If we include finished goods, then turns are TH/(WIP + FGI). But Little's law still applies, so this ratio represents the inverse of the total average time for a job to traverse the line plus the finished goods crib. Hence, intuitively, inventory turns are one divided by the average residence time of inventory in the system.

In a sense, Little's law is the "$F = ma$" of factory physics. It is a broadly applicable equation that relates three fundamental quantities. At the same time, Little's law can be viewed as a truism about units. It merely indicates the obvious fact that we can measure WIP level in a station, line, or system in units of jobs or time. For instance, a line that produces 100 crankcases per day and has a WIP level of 500 crankcases has five days of WIP in it. Little's law is a statement that this unit's conversion is valid for average WIP, cycle time, and throughput, or

$$\text{CT} = \frac{\text{WIP}}{\text{TH}}$$

or

$$5 \text{ days} = \frac{500 \text{ crankcases}}{100 \text{ crankcases per day}}$$

We can now generalize the results shown in Table 7.3 and Figure 7.4 to achieve our original objective of giving a precise summary of the relationship between WIP and throughput for a "best-case" (i.e., zero-variability) line. We can then apply Little's law to extend this to describe the relationship between WIP and cycle time. Since these relationships were derived for perfect lines with no variability, the following expressions indicate the *maximum throughput* and *minimum cycle time* for a given WIP level for any system having parameters r_b and T_0. The resulting equations are our next *Factory Physics* law.

Law (Best-Case Performance): *The minimum cycle time for a given WIP level w is given by*

$$\mathrm{CT}_{\mathrm{best}} = \begin{cases} T_0 & \text{if } w \leq W_0 \\ \dfrac{w}{r_b} & \text{otherwise} \end{cases}$$

The maximum throughput for a given WIP level w is given by

$$\mathrm{TH}_{\mathrm{best}} = \begin{cases} \dfrac{w}{T_0} & \text{if } w \leq W_0 \\ r_b & \text{otherwise} \end{cases}$$

One conclusion we can draw from this is that, contrary to the popular slogan, zero inventory is *not* a realistic goal. Even under perfect deterministic conditions, zero inventory yields zero throughput and therefore zero revenue. A more realistic "ideal" WIP is the critical WIP W_0.

Penny Fab One represents an ideal (zero-variability) situation, in which it is optimal to maintain a WIP level equal to the number of machines. Of course, in the real world there are not many factories that run with such low WIP levels. Indeed, in many production lines the WIP-to-machines ratio is closer to 20:1 (Bradt 1983). If this ratio were to hold for Penny Fab One, the cycle time would be almost seven days with 80 jobs in WIP. Obviously, this is much worse than a cycle time of eight hours at a WIP level of four jobs (i.e., the "optimal" level). Why, then, do actual plants operate so far from the ideal of the critical WIP level?

Unfortunately, Little's law offers little help. Since TH = WIP/CT, we can have the same throughput with large WIP levels and long cycle times, or with low WIP levels and short cycle times. The problem is that Little's law is only one relation among three quantities. We need a second relation if we are to uniquely determine two quantities, given the third (e.g., predict both WIP and cycle time from throughput). Sadly, there is no universally applicable second relationship among WIP, cycle time, and throughput. The best we can do is to characterize the behavior of a line under specific assumptions. In addition to the best case, which we considered above, we will treat two other scenarios, which we term the **worst case** and the **practical worst case.**

7.3.2 Worst-Case Performance

Instead of imagining the best possible behavior of a line, we consider the worst. Specifically, we seek the *maximum cycle time* and *minimum throughput* possible for a line with bottleneck rate r_b and raw process time T_0. This will enable us to bracket the behavior and gauge the performance of real lines. If a line is closer to the worst case than to the best case, then there are some real problems (or opportunities, depending on your perspective).

To facilitate our discussion of the worst case, recall that we are assuming a constant amount of work is maintained in the line at all times. Whenever a job finishes, another is started. One way that this could be achieved in practice would be to transport jobs through the line on *pallets*. Whenever a job is finished, it is removed from its pallet and the pallet immediately returns to the front of the line to carry a new job. The WIP level, therefore, is equal to the (fixed) number of pallets.

Now, imagine yourself sitting on a pallet riding around and around a best-case line with WIP equal to the critical WIP (e.g., Penny Fab One with four jobs). Each time you arrive at a station, a machine is available to begin work on the job immediately. It is precisely because there is no waiting (queueing) that this line achieves the minimum possible cycle time of T_0.

To get the longest possible cycle times for this system, we must somehow increase the waiting time without changing the *average* processing times (otherwise we would change r_b and T_0). The very worst we could possibly make waiting time would be that every time our pallet reached a station, we found ourselves waiting behind *every* other job in the line. How could this possibly occur?

Consider the following. Suppose that you are riding on pallet number 4 in a modified Penny Fab One with four pallets. However, instead of all jobs requiring exactly two hours at each station, suppose that jobs on pallet 1 require eight hours, while jobs on pallets 2, 3, and 4 require zero hours. The average processing time at each station is

$$\frac{8+0+0+0}{4} = 2 \text{ hours}$$

as before, and hence we still have $r_b = 0.5$ job per hour and $T_0 = 8$ hours. However, every time your pallet reaches a station, you find pallets 1, 2, and 3 ahead of you (see Figure 7.5). The slow job on pallet 1 causes all the other jobs to pile up behind it at all times. This is the absolute maximum amount of waiting time it is possible to introduce, and hence this represents the worst case.

The cycle time for this system is

$$8 + 8 + 8 + 8 = 32 \text{ hours}$$

or $4T_0$, and since four jobs are output each time pallet 1 finishes on station 4, the throughput is

$$\frac{4}{32} = \frac{1}{8} \text{ job per hour}$$

or $1/T_0$ jobs per hour. Notice that the product of throughput and cycle time is $\frac{1}{8} \times 32 = 4$, which is the WIP level, so, as always, Little's law holds.

Let us summarize these results for a general line as our next factory physics law.

Law (Worst-Case Performance): *The worst-case cycle time for a given WIP level w is given by*

$$\text{CT}_{\text{worst}} = wT_0$$

The worst-case throughput for a given WIP level w is given by

$$\text{TH}_{\text{worst}} = \frac{1}{T_0}$$

It is interesting to note that both the best-case and worst-case performances occur in systems with no randomness. There is *variability* in the worst-case system, since jobs have different process times; but there is no *randomness,* since all process times are completely predictable. The literature on quality management stresses the need